



US011263399B2

(12) **United States Patent**
Ivan et al.

(10) **Patent No.: US 11,263,399 B2**
(45) **Date of Patent: Mar. 1, 2022**

(54) **CORRECTING INPUT BASED ON USER CONTEXT**

(71) Applicant: **Apple Inc.**, Cupertino, CA (US)

(72) Inventors: **Jason J. Ivan**, Gilroy, CA (US);
Christopher S. Linn, Palo Alto, CA (US); **Douglas R. Davidson**, Palo Alto, CA (US)

(73) Assignee: **Apple Inc.**, Cupertino, CA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 170 days.

(21) Appl. No.: **15/702,587**

(22) Filed: **Sep. 12, 2017**

(65) **Prior Publication Data**

US 2019/0034405 A1 Jan. 31, 2019

Related U.S. Application Data

(60) Provisional application No. 62/538,979, filed on Jul. 31, 2017.

(51) **Int. Cl.**

G06F 3/048 (2013.01)
G06F 40/274 (2020.01)
G06F 16/33 (2019.01)
G06F 40/211 (2020.01)
G06F 40/242 (2020.01)

(Continued)

(52) **U.S. Cl.**

CPC **G06F 40/274** (2020.01); **G06F 16/3344** (2019.01); **G06F 40/211** (2020.01); **G06F 40/242** (2020.01); **G06F 40/279** (2020.01); **G06F 3/04842** (2013.01)

(58) **Field of Classification Search**

CPC G06F 3/0481–0489; G06F 40/274; G06F

40/211; G06F 40/242; G06F 40/279; G06F 16/3344; G06F 3/04842; G06F 40/232; G06F 40/166; G06F 40/58; G06F 3/0237; G06F 16/3322; G06F 16/90324; G06F 40/20; G06F 3/048

See application file for complete search history.

(56)

References Cited

U.S. PATENT DOCUMENTS

5,875,443 A * 2/1999 Nielsen G06F 40/242
7,100,123 B1 * 8/2006 Todd G06F 16/30
715/862

(Continued)

OTHER PUBLICATIONS

Flor, “Four Types of Context for Automatic Spelling Correction,” Educational Testing Service, <https://atala.org/IMG/pdf/Flor-TAL53-3.pdf>, 39 pages.

Primary Examiner — Steven B Theriault

(74) *Attorney, Agent, or Firm* — Fletcher Yoder PC

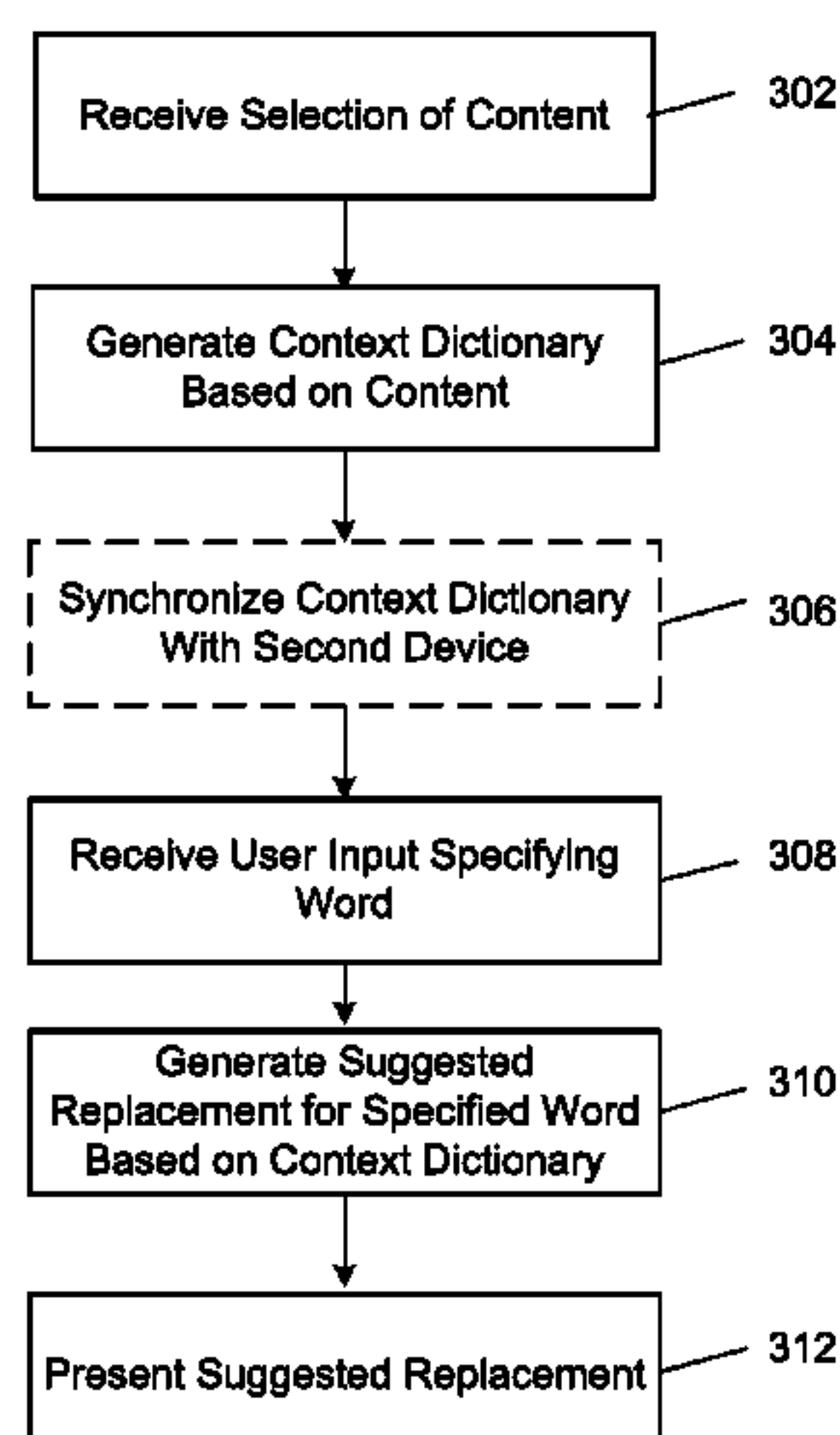
(57)

ABSTRACT

In some implementations, a computing device can generate user input correction suggestions based on the user’s context. For example, the user’s context can include content that the user has open or has recently opened on the computing device or another computing device. For example, when the user opens an item of content, the computing device can generate a context dictionary that includes words, phrases, etc., that describe the opened content. When the user provides input (e.g., text, speech, etc.) the computing device can use the context dictionary to generate input correction suggestions. The computing device can synchronize the context dictionary with other computing devices that the user may be using so that the user’s context on one device can be used by another device to generate input correction suggestions.

24 Claims, 5 Drawing Sheets

300
↓



Page 2

2009/0094221	A1 *	4/2009	Cameron	G06F 16/3322
2009/0216690	A1 *	8/2009	Badger	H04M 1/2748 706/11
2009/0241104	A1 *	9/2009	Amiga	G06F 8/71 717/174
2009/0288037	A1 *	11/2009	Lawton	G06F 3/0481 715/804
2010/0114887	A1 *	5/2010	Conway	G06F 16/374 707/737
2010/0122194	A1 *	5/2010	Rogers	G06F 3/04817 715/769
2010/0161733	A1 *	6/2010	Bower	G06F 3/0237 709/206
2010/0241663	A1 *	9/2010	Huang	G06Q 30/02 707/770
2010/0275118	A1 *	10/2010	Iakobashvili	G06F 40/232 715/257
2011/0014944	A1 *	1/2011	Samoylov	H04M 1/72522 455/556.1
2011/0167350	A1 *	7/2011	Hoellwarth	G06F 3/0483 715/727
2011/0201387	A1 *	8/2011	Paek	G06F 40/166 455/566
2011/0202836	A1 *	8/2011	Badger	G06F 3/0237 715/702
2011/0246575	A1 *	10/2011	Murayama	G06F 16/93 709/204
2012/0046068	A1 *	2/2012	Katpelly	H04W 4/029 455/550.1
2012/0072867	A1 *	3/2012	Schlegel	G06F 3/0481 715/808
2012/0089400	A1 *	4/2012	Henton	G10L 13/02 704/260
2012/0117499	A1 *	5/2012	Mori	H04M 1/72569 715/765
2012/0150533	A1 *	6/2012	O'Neill	G06F 16/24578 704/9
2012/0173222	A1 *	7/2012	Wang	G06F 17/273 704/2
2012/0203541	A1 *	8/2012	Liu	G06F 3/0236 704/8
2012/0233567	A1 *	9/2012	Brown	G06Q 30/0601 715/810
2012/0296635	A1 *	11/2012	Brockett	G06F 40/131 704/9
2012/0303371	A1 *	11/2012	Labsky	G10L 17/00 704/260
2012/0304124	A1 *	11/2012	Chen	G06F 3/0236 715/825
2012/0330669	A1 *	12/2012	Narayanan	G10L 21/06 704/271
2013/0046544	A1 *	2/2013	Kay	G06F 3/04883 704/275
2013/0108115	A1 *	5/2013	Hwang	G06K 9/033 382/106
2013/0159847	A1 *	6/2013	Banke	G06F 17/2735 715/259
2013/0275120	A1 *	10/2013	DeGross	G06F 40/40 704/9
2013/0332822	A1	12/2013	Willmore et al.	
2014/0035823	A1	2/2014	Khoe et al.	
2014/0067371	A1	3/2014	Liensberger	
2014/0089947	A1 *	3/2014	Han	G06F 9/44505 719/328
2014/0104175	A1 *	4/2014	Ouyang	G06F 17/2735 345/168
2014/0163954	A1 *	6/2014	Joshi	G06F 40/274 704/9
2014/0244239	A1 *	8/2014	Nicholson	G06F 17/2735 704/9
2014/0281943	A1 *	9/2014	Prilepov	G06F 40/242 715/257
2014/0317495	A1 *	10/2014	Lau	G06F 17/273 715/257
2014/0354550	A1 *	12/2014	Mendonca	G06F 3/02 345/168

(56)

References Cited

U.S. PATENT DOCUMENTS

2015/0067486 A1* 3/2015 Hochman G06F 17/273 715/257

2015/0067491 A1* 3/2015 Rajamony G06F 40/274 715/271

2015/0088493 A1* 3/2015 Thakore G06F 40/242 704/10

2015/0100301 A1* 4/2015 Phadke G06F 40/47 704/4

2015/0127346 A1* 5/2015 Gruenstein G10L 15/08 704/254

2015/0186347 A1* 7/2015 Taniguchi G06F 40/166 715/256

2015/0309667 A1* 10/2015 Wang G06F 9/453 715/781

2015/0309984 A1* 10/2015 Bradford G06F 40/263 704/8

2015/0331855 A1* 11/2015 Rylov G06F 17/28 704/2

2016/0048270 A1* 2/2016 An G06F 3/04886 715/773

2016/0110327 A1* 4/2016 Knox G06F 17/273 704/9

2016/0196150 A1* 7/2016 Jing G06F 9/453 715/708

2016/0224540 A1* 8/2016 Stewart G06F 17/273

2016/0267128 A1* 9/2016 Dumoulin G06F 16/3322

2016/0357730 A1* 12/2016 Eberbach G06F 17/2735

2016/0366534 A1* 12/2016 Griesmann H04L 67/42

2017/0052659 A1* 2/2017 Ivanov G06F 3/0482

2017/0098159 A1* 4/2017 Sharifi G06F 3/0484

2017/0103057 A1* 4/2017 Guerra G06F 17/2735

2017/0116174 A1* 4/2017 Peterson G06F 17/276

2017/0220536 A1* 8/2017 Chiba G06F 40/253

2017/0249017 A1* 8/2017 Ryu G06F 17/276

2018/0032499 A1* 2/2018 Hampson G06F 17/273

2018/0040020 A1* 2/2018 Kurian G06F 16/9535

2018/0143956 A1* 5/2018 Skarbovsky G10L 15/26

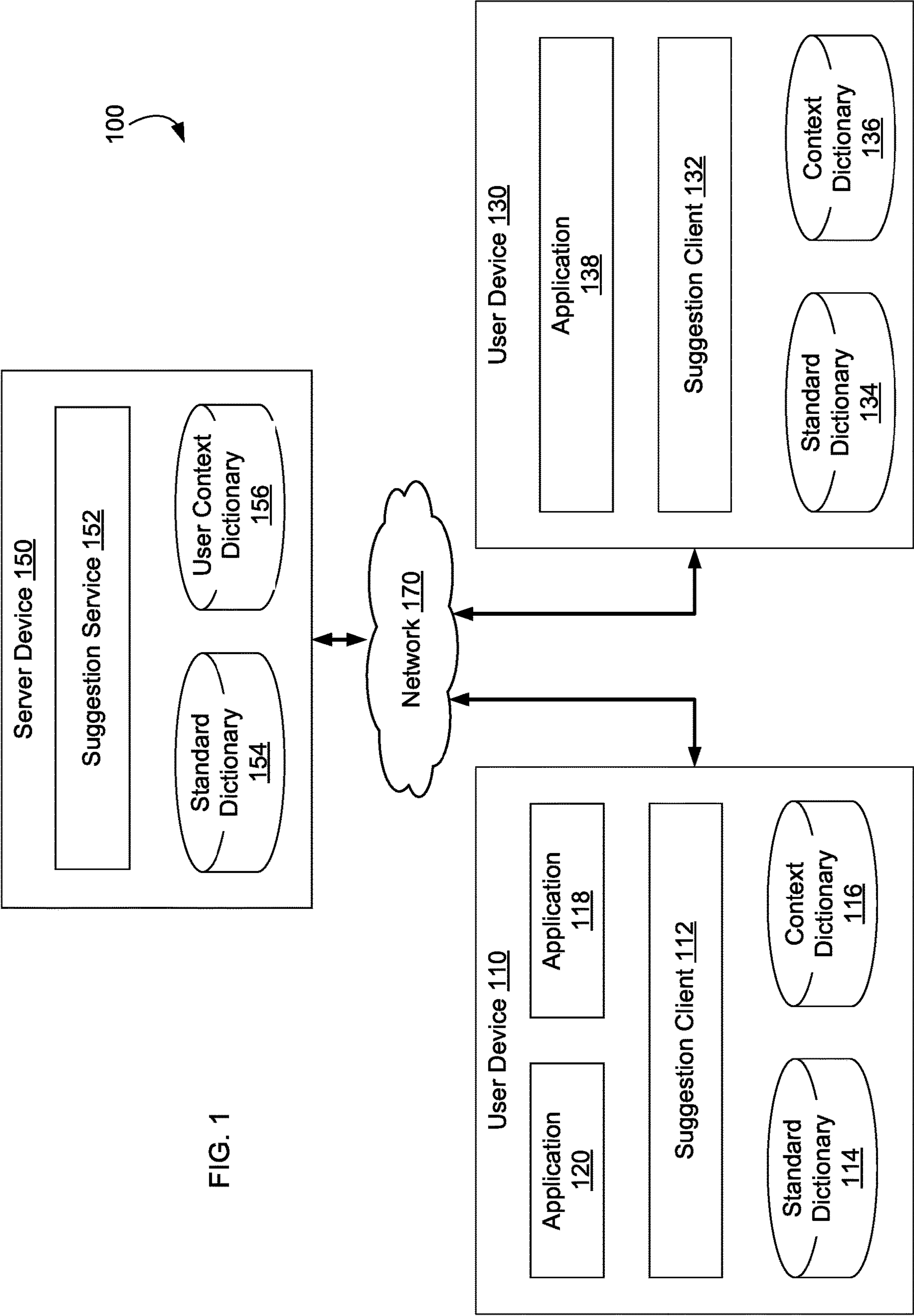
2018/0143970 A1* 5/2018 Skarbovsky G10L 15/26

2018/0143974 A1* 5/2018 Skarbovsky G06F 40/58

2018/0144747 A1* 5/2018 Skarbovsky G06F 40/232

2019/0018584 A1* 1/2019 Jung G06F 3/04883

* cited by examiner



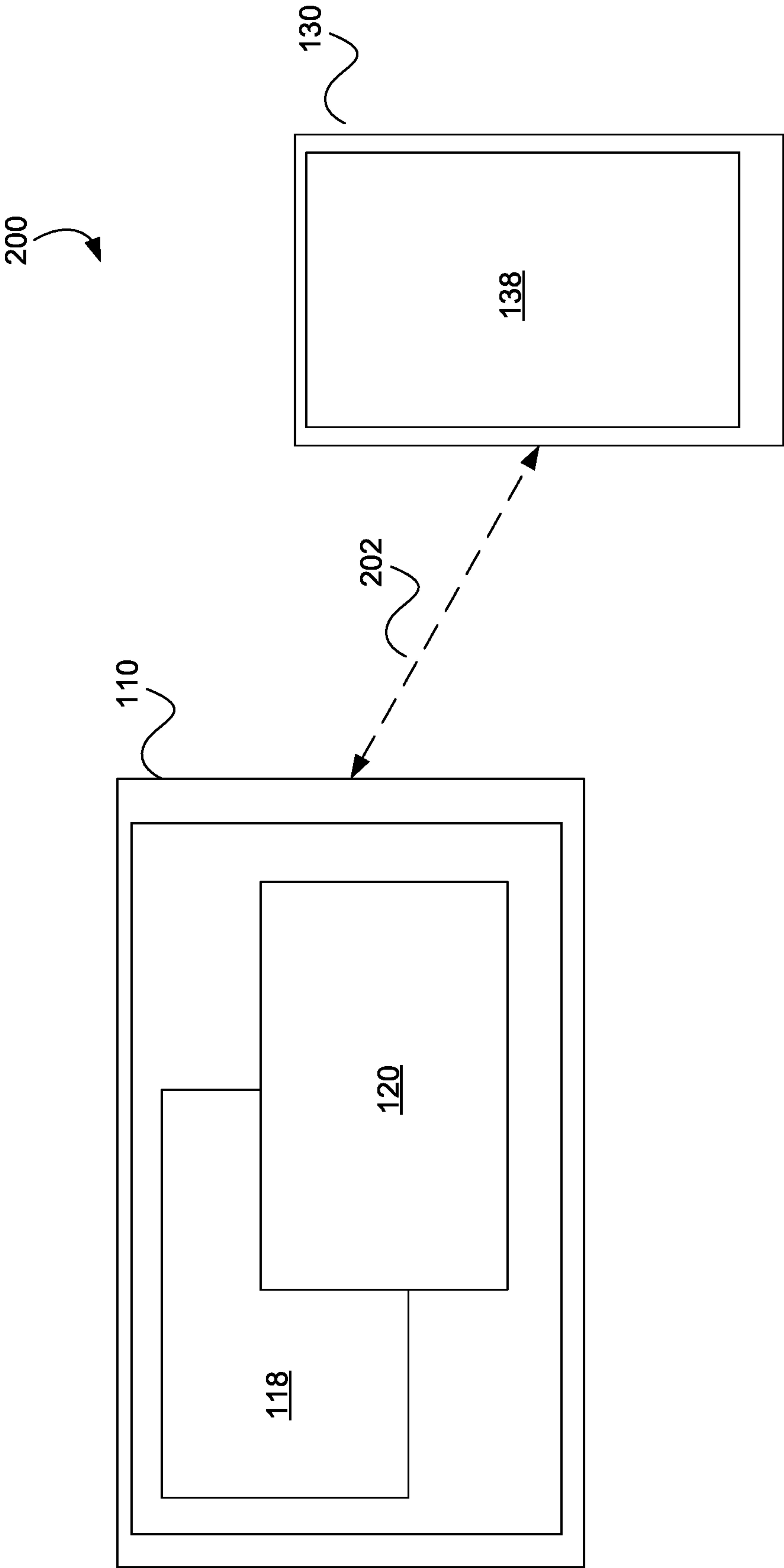


FIG. 2

FIG. 3

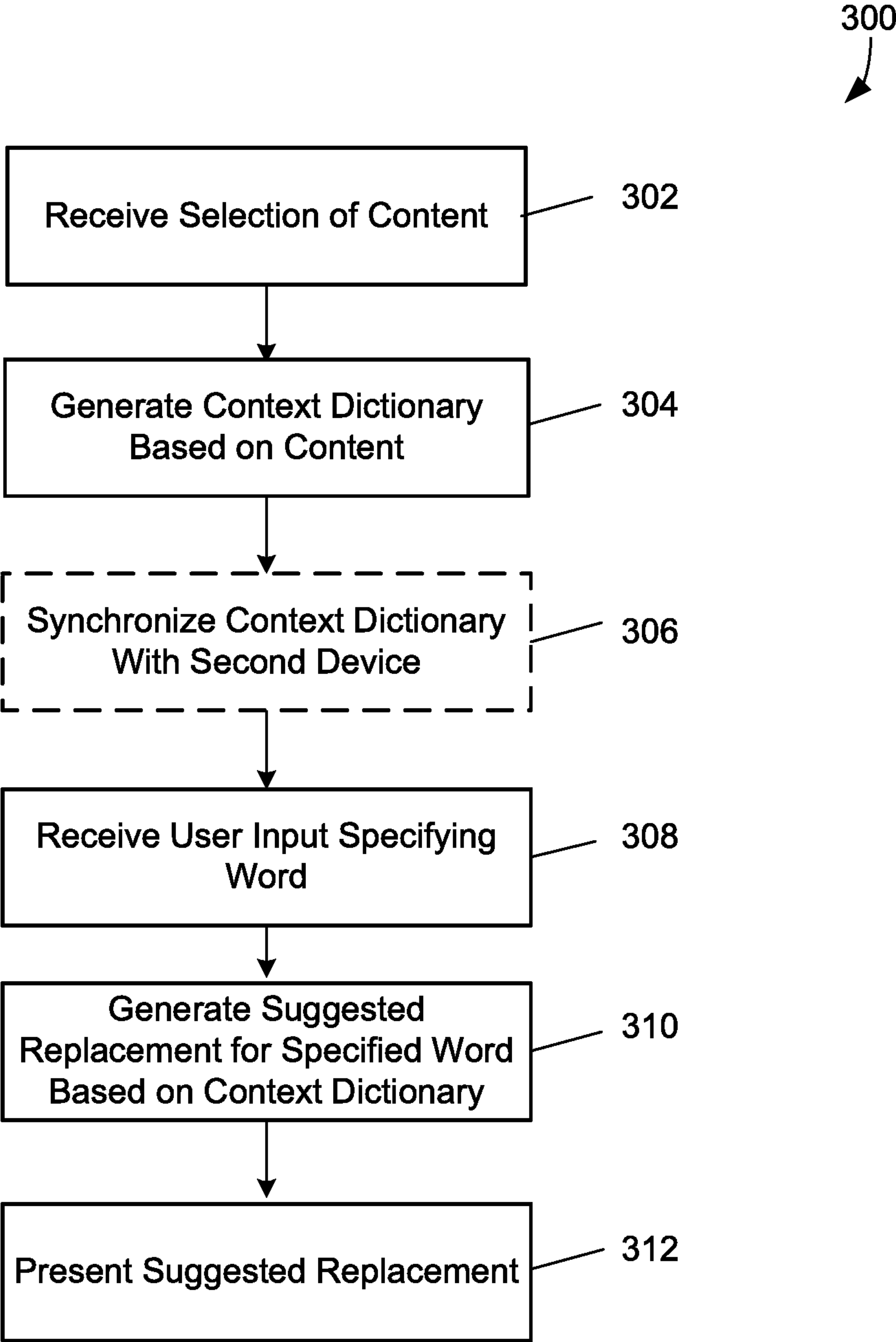
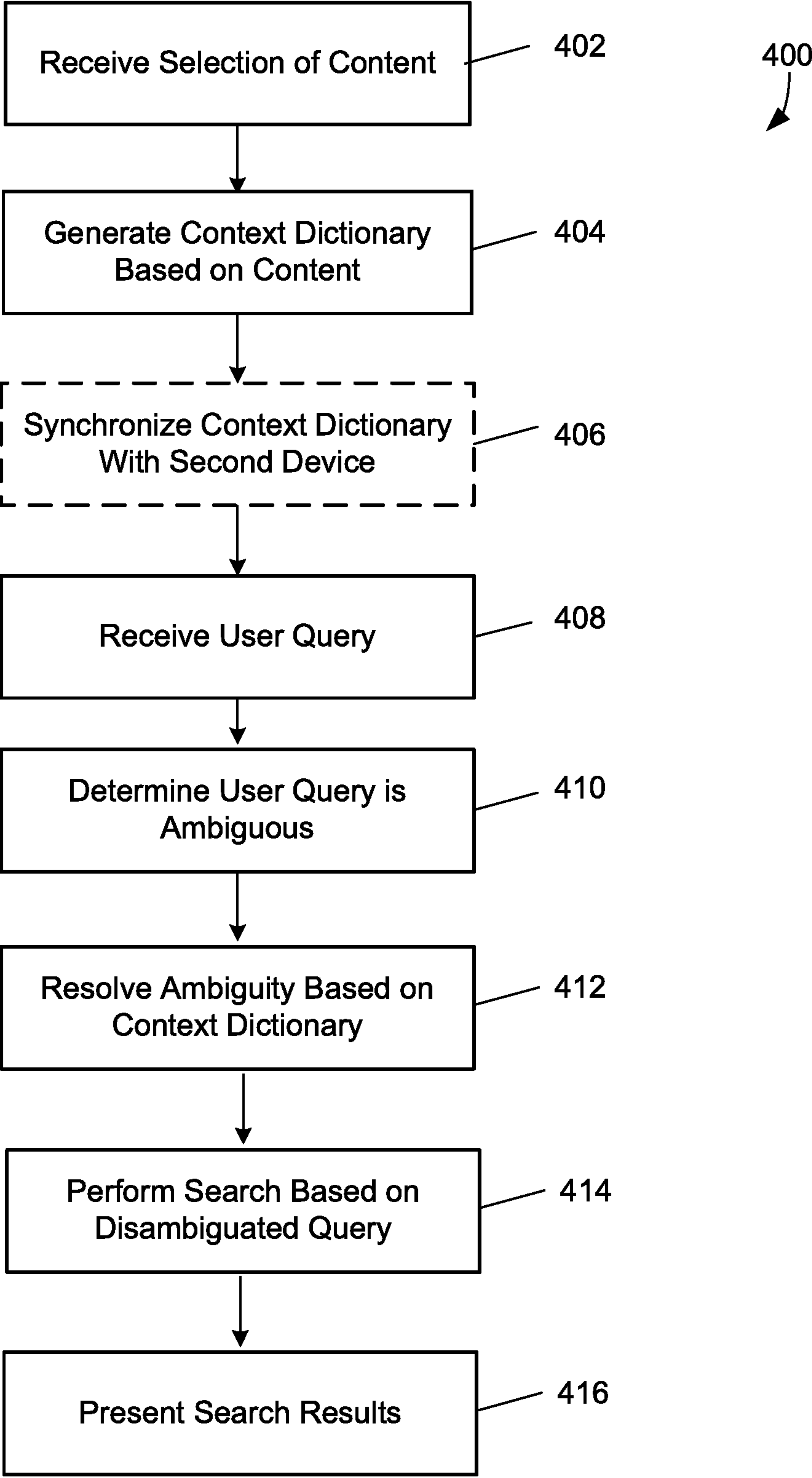


FIG. 4



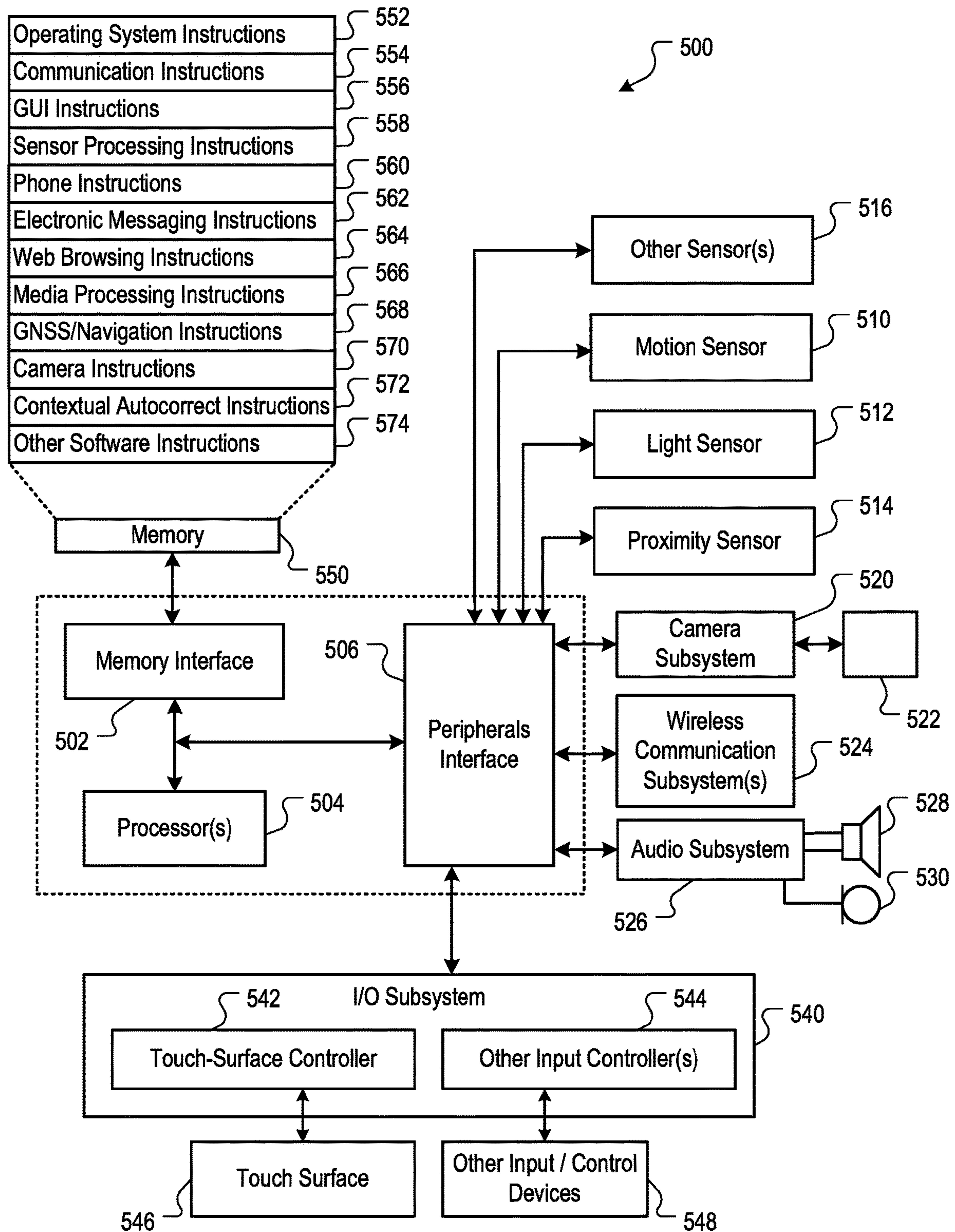


FIG. 5

1

CORRECTING INPUT BASED ON USER
CONTEXT

RELATED APPLICATION

This application claims priority to U.S. Provisional Application No. 62/538,979, filed on Jul. 31, 2017, the content of which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

The disclosure generally relates to generating suggestions for correcting user input.

BACKGROUND

User input correction mechanisms are commonplace on modern computing devices. For example, a computing device can receive textual input, detect a spelling error in a word or grammatical error in a phrase or a sentence and recommend or suggest a correction. However, these input correction technologies do not take into account the user's or device's context when determining and/or making a suggestion for a correction.

SUMMARY

In some implementations, a computing device can generate user input correction suggestions based on the user's context. For example, the user's context can include content that the user has open or has recently opened on the computing device or another computing device. For example, when the user opens an item of content, the computing device can generate a context dictionary that includes words, phrases, etc., that describe the opened content. When the user provides input (e.g., text, speech, etc.) the computing device can use the context dictionary to generate input correction suggestions. The computing device can synchronize the context dictionary with other computing devices that the user may be using so that the user's context on one device can be used by another device to generate input correction suggestions.

Particular implementations provide at least the following advantages. The computing device can provide correction suggestions that are relevant to the user's current activities on one or more of the user's computing devices.

Details of one or more implementations are set forth in the accompanying drawings and the description below. Other features, aspects, and potential advantages will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a block diagram of an example system for automatically correcting user input based on user context.

FIG. 2 is an example illustration depicting graphical user interfaces of two user devices.

FIG. 3 is flow diagram of an example process for generating correction suggestions based on user context.

FIG. 4 is flow diagram of an example process for disambiguating user input based on user context.

FIG. 5 is a block diagram of an example computing device that can implement the features and processes of FIGS. 1-4.

Like reference symbols in the various drawings indicate like elements.

2

DETAILED DESCRIPTION

FIG. 1 is a block diagram of an example system 100 for automatically correcting user input based on user context.

For example, system 100 can suggest corrections to user input based on content (e.g., word processing document, email message, text message, contact card, webpage, etc.) that the user is viewing and/or has recently viewed on one or more of the user's devices. The selected and/or recently viewed content can provide context clues that allow system 100 to suggest corrections to the user's input.

In some implementations, system 100 can include user device 110. For example, user device 110 can be a laptop computer, desktop computer, tablet computer, smartphone, smart watch, in-car infotainment system, and/or other computing device.

In some implementations, user device 110 can include suggestion client 112. For example, suggestion client 112 can collect context data (e.g., text describing selected and/or viewed content) and/or generate suggestions for correcting user input based on the collected context data.

In some implementations, suggestion client 112 can provide user input correction functionality based on a default or standard dictionary 114. For example, standard dictionary 114 can include words, phrases, grammatical rules, etc., for one or more languages used by user device 110. When a user provides input (e.g., textual input, voice input, etc.), suggestion client 112 can suggest alternatives to the user's input based on the data in standard dictionary 114. For example, if the user enters text input that misspells a word, suggestion client 112 can determine a correct replacement word for the misspelled word based on standard dictionary 114. Standard dictionary 114 can include the same data across devices, for example.

In some implementations, suggestion client 112 can provide user input correction functionality based on context dictionary 116. For example, context dictionary 116 can be dynamically generated by suggestion client 112. When a user selects content to view on user device 110, suggestion client 112 can determine, based on the selected content, words, phrases, ideas, etc., that describe the content the user has selected to view. For example, suggestion client 112 can parse the text of the content and/or content metadata to determine words, phrases, ideas, etc., described in the content. Suggestion client 112 can generate entries in context dictionary 116 that includes the words, phrases, ideas, etc., parsed from the selected content. Each context dictionary entry can include, for example, a word from the selected content, a timestamp indicating when the word was added to the context dictionary, and/or a timestamp indicating when the corresponding content item was closed (e.g., is no longer being accessed by the user). When a user provides input (e.g., textual input, voice input, etc.), suggestion client 112 can suggest alternatives to the user's input based on the data in context dictionary 116. For example, if the user enters text input that misspells a word, suggestion client 112 can determine a correct replacement word for the misspelled word based on context dictionary 114. If the user enters ambiguous input, suggestion client 112 can disambiguate the input based on the data stored in context dictionary 116.

In some implementations, suggestion client 112 can prevent a standard dictionary correction based on context dictionary 114. For example, a user may enter text corresponding to a word that is not included in standard dictionary 114. Suggestion client 112 may determine that the entered word includes an error, is unknown, or misspelled because the word is not included in standard dictionary 114.

and cause user device 110 to present an indication that the entered word is misspelled and/or present suggested corrections for the entered word. However, in some cases, the entered word may correspond to a term, person's name, or other type of word not included in standard dictionary 114 but included in content the user has recently selected to view on user device 110 (or some other user device, such as user device 130). In addition to comparing the entered word to entries in standard dictionary 114, suggestion client 112 may compare the entered word to entries in context dictionary 116. When suggestion client 112 finds a context entry in context dictionary 116 that matches the entered word, suggestion client 112 can determine that, even though the word is not in standard dictionary 114, the entered word is not an error or misspelled word because the entered word is in context dictionary 116. When the entered word is found in context dictionary 116, suggestion client 112 can prevent a user device 110 from presenting an indication that the entered word is misspelled and/or presenting suggested corrections for the entered word.

In some implementations, suggestion client 112 can ignore or remove entries in context dictionary when a period of time elapses. For example, context dictionary entries that do not have a timestamp indicating that the corresponding content item is closed can be considered active context entries. Context dictionary entries that have a timestamp indicating that the corresponding content item is closed and where the amount of time that has elapsed since the content item was closed is less than the configured period of time can be considered active context entries. Context dictionary entries that have a timestamp indicating that the corresponding content item is closed and where the amount of time that has elapsed since the content item was closed is greater than the configured period of time can be considered inactive context entries. When generating suggested replacement words or input, suggestion client 112 can use the active context entries in context dictionary 116 to generate the replacement input suggestions. Suggestion client 112 can ignore or remove inactive context entries in context dictionary 116. Thus, suggestion client 112 can generate suggestions based on currently open content items and/or recently closed content items.

In some implementations, context dictionary 116 can be synchronized between devices associated with the same user so that content selected by the user on one device can influence alternative word suggestions on another user device. For example, system 100 can include user device 130. User device 130 can be configured similarly to user device 110. For example, user device 130 can include suggestion client 132 corresponding to suggestion client 112. User device 130 can include standard dictionary 134 corresponding to standard dictionary 114. User device 130 can include context dictionary 136 corresponding to context dictionary 116. When suggestion client 132 modifies context dictionary 136 on user device 130 based on content selected by the user on user device 130, suggestion client 132 can synchronize context dictionary 136 with context dictionary 116 on use device 110 so that each user device has the same context dictionary. Thus, when suggestion client 112 suggests an alternative word in response to receiving user input on user device 110, suggestion client 112 can generate the suggestion based on content selected by the user on user device 130.

In some implementations, system 100 can include server device 150. For example, server device 150 can represent a computing device or multiple computing devices configured to provide a network based suggestion service 152. Like user

device 110 and user device 130, server device 150 can include a standard dictionary 154 and a user context dictionary 156. For example, when user device 110 and/or user device 130 updates their respective context dictionaries 116/136, user device 110 and/or user device 130 can synchronize their context dictionaries with user context dictionary 156 on server device 150 through network 170 (e.g., local area network, wide area network, wireless network, the Internet, etc.). In some implementations, the synchronization of context dictionary 116 on user device 110 and context dictionary 136 on user device 130 can be performed through user context dictionary 156 on server device 150. In some implementations, user device 130 and user device 110 can synchronize their context dictionaries directly without using server device 150.

In some implementations, suggestion service 152 can provide user input correction suggestions to user device 110 and/or user device 130. For example, when a user selects and/or opens a content item on user device 130, suggestion client 132 can update context dictionary 136 based on the content of the content item, as described above. Suggestion client 132 can then synchronize content dictionary 136 with user context dictionary 156 on server device 150. Similarly, when the user selects and/or opens a content item on user device 110, suggestion client 112 can update context dictionary 116 based on the content of the content item, as described above. Suggestion client 112 can then synchronize content dictionary 116 with user context dictionary 156 on server device 150 through network 170 (e.g., the Internet). When the user provides input (e.g., text input, voice input, etc.) to user device 110, suggestion client 112 can send the user input to suggestion service 152. Suggestion service 152 can then generate suggestions for correcting the user input and send the suggestions to suggestion client 112. Suggestion client 112 can then present the suggested corrections on a display of user device 110 so that the user can select a suggested correction. In some implementations, suggestion client 112 can automatically change the user input to automatically correct the user input according to a correction suggested by suggestion service 152. Thus, suggestion client 112 can generate suggested corrections locally at user device 110 and/or receive suggested corrections from suggestion service 152 on server device 150.

In some implementations, user device 110 and/or user device 130 can include applications, utilities, operating systems, or other software functions. For example, these software functions are represented by applications 118, 120 for user device 110 and application 138 for user device 130. For example, applications 118, 120, and 128 can correspond to word processing applications, web browsers, web applications, a contacts application (e.g., address book), electronic mail applications, electronic messaging applications, social media applications, and/or any other type of application that can be used by the user to select and/or view content. These applications can interact with the local suggestion client (e.g., suggestion client 112 for user device 110, suggestion client 132 for user device 130) to send the suggestion client content selected by the user so that the suggestion client can generate the local context dictionary. These applications can interact with the local suggestion client to receive user input correction suggestions from the suggestion client based on the content recently selected by the user.

FIG. 2 is an example illustration 200 depicting graphical user interfaces of two user devices. Illustration 200 will be used to describe various use cases for system 100 described above. For example, illustration 200 includes user device

5

110 and user device 130, as described above with respect to system 100. Illustration 200 also includes applications 118 and 120 on user device 110 and application 138 on user device 130. The synchronization of context libraries between user device 110 and user device 130 is represented by dashed line 202. As described above, the synchronization of context libraries can be performed directly between user device 110 and user device 130 or through server device 150. While some of the use cases described below are described with reference to operations performed on a single device (e.g., user device 110), similar operations may be distributed across multiple devices (e.g., user device 110 and user device 130). Similarly, where use cases describe operations performed across multiple devices, similar operations can be performed within a single device.

Autocorrect Based on Context

In a first example use case, application 118 can correspond to a web browser and application 120 can correspond to a word processing application. The user may interact with web browser 118 to do research on bears. While viewing a website describing bears, web browser 118 can send the website content to suggestion client 112. Suggestion client 112 can generate context dictionary entries based on the content of the website describing bears. While the website is still open or shortly after closing browser application 118, the user types text, including the word “bares”, into word processing application 120. Word processing application 120 communicates the textual user input to suggestion client 112. Suggestion client 112 determines, based on context dictionary 116, that the user was recently looking at content describing “bears” and generates a correction suggestion that indicates the word “bares” should be corrected to the word “bears”. Suggestion client 112 can then send the correction suggestion to word processing application 120 where application 120 can replace the word “bares” with the word “bears.”

In some implementations, the correction suggestion can include a graphic to be presented instead of or in addition to a word. For example, rather than suggesting that the word “bares” be corrected with the word “bears,” suggestion client 112 can suggest that the word “bares” be replaced with the word “bears” and/or a graphic (e.g., emoji, icon, picture, etc.) representing a bear.

Autocorrect in Email

In a second example use case, application 138 can correspond to an email application. For example, the user may receive an email from a person named “Leezah” or she may be mentioned in an email thread or on the CC list but is not in the user’s address book or contacts application. When the email is opened or selected by the user, email application 138 can send the content of the email message and/or message thread to suggestion client 132. Suggestion client 132 can generate context dictionary entries on context dictionary 136 based on the content of the email message or thread, including the name “Leezah”. When the user provides input to respond to the selected message or message thread, email application 138 may send the message thread to suggestion client 132 as user input to be analyzed and/or corrected. Normally, suggestion client 132 would identify “Leezah” as a misspelled word. However, since the word “Leezah” is within the context dictionary 136, suggestion client 132 can determine that “Leezah” is not a spelling mistake and suggestion client 132 will not suggest a correction for the word “Leezah”.

6

Later, when the user is typing the email response, the user may enter the word “Liza.” This input can be sent by email application 138 to suggestion client 132. Suggestion client 132 can analyze context dictionary 136, determine that “Liza” is not an entry in context dictionary 136 but “Leezah” is. Suggestion client 132 can then generate a correction suggestion suggesting that the input “Liza” be replaced by “Leezah” and send the correction suggestion to email application 138. Email application 138 can then present the correction suggestion to the user and the user can provide input selecting the word “Leezah” as a replacement for the word “Liza.”

Autocorrect Across Devices

In a third example use case and as a variation on the example email use case above, the user may store an address book or contacts application on one device but be responding to an email message on another device. For example, application 138 on user device 130 can correspond to the email application and application 118 on user device 110 can correspond to the address book application. The user may look up the email address for a contact named “Leezah” in application 118. Address book application 118 can send the content of the contact information for Leezah to suggestion client 112. Suggestion client 112 can generate context entries on context dictionary 116 for the contact information for Leezah, including the word “Leezah”. As described above, context dictionary 116 can be synchronized with context dictionary 136 so that context dictionary 136 includes the context entries corresponding to the contact information for Leezah. When the user provides input to respond to the selected message or message thread, email application 138 may send the message thread to suggestion client 132 as user input to be analyzed and/or corrected. Normally, suggestion client 132 would identify “Leezah” as a misspelled word. However, since the word “Leezah” is within the context dictionary 136, suggestion client 132 can determine that “Leezah” is not a spelling mistake and suggestion client 132 will not suggest a correction for the word “Leezah”.

Later, when the user is typing the email response, the user may enter the word “Liza.” This input can be sent by email application 138 to suggestion client 132. Suggestion client 132 can analyze context dictionary 136, determine that “Liza” is not an entry in context dictionary 136 but “Leezah” is. Suggestion client 132 can then generate a correction suggestion suggesting that the input “Liza” be replaced by “Leezah” and send the correction suggestion to email application 138. Email application 138 can then present the correction suggestion to the user and the user can provide input selecting the word “Leezah” as a replacement for the word “Liza.”

Voice Command Disambiguation

In a fourth example use case, application 138 can correspond to a voice assistant artificial intelligence application. The user may speak a command to application 138, such as “how are the Giants doing?” If there is an ambiguity in the command (e.g., currently there is a baseball season and football season overlap), application 138 may need to follow up with a clarifying question to determine whether the user wants to know about the Giants baseball team or the Giants football team. However, if the user has recently viewed a webpage related to Giants baseball, context dictionary 136 will have context entries related to Giants baseball. When

the user asks “how are the Giants doing?”, application 138 can send a text version of the user’s voice input to suggestion client 132. Suggestion client 132 can determine that the user has recently viewed a webpage related to Giant’s baseball and generate a correction suggestion indicating that “Giants” refers to the Giants baseball team and send the correction suggestion to application 138. Application 138 can then formulate a response to the user’s request by determining how the Giants baseball team is doing in the current baseball season without asking the user for clarification as to whether the user meant the Giants baseball team or the Giants football team.

Example Processes

To enable the reader to obtain a clear understanding of the technological concepts described herein, the following processes describe specific steps performed in a specific order. However, one or more of the steps of a particular process may be rearranged and/or omitted while remaining within the contemplated scope of the technology disclosed herein. Moreover, different processes, and/or steps thereof, may be combined, recombined, rearranged, omitted, and/or executed in parallel to create different process flows that are also within the contemplated scope of the technology disclosed herein. Additionally, while the processes below may omit or briefly summarize some of the details of the technologies disclosed herein for clarity, the details described in the paragraphs above may be combined with the process steps described below to get a more complete and comprehensive understanding of these processes and the technologies disclosed herein.

FIG. 3 is flow diagram of an example process 300 for generating correction suggestions based on user context. For example, process 300 can be performed by user device 110 to correct user input based on content that the user has recently selected to view.

At step 302, user device 110 can receive a selection of content. For example, user device 110 can receive user input selecting a webpage, contact information, word processing document, slideshow presentation, or any other type of content. The content can be selected by interacting with an application (e.g., web browser, address book application, word processing document, etc.) running on user device 110.

At step 304, user device 110 can generate a context dictionary based on the selected content. For example, the application (e.g., application 118, application 120) presenting the selected content can send the selected content to suggestion client 112. Suggestion client 112 can parse the selected content and generate context entries in context dictionary 116.

At optional step 306, user device 110 can synchronize the context dictionary with a second device. For example and as described above, user device 110 can synchronize context dictionary 116 with context dictionaries managed by user device 130 and/or server device 150. The synchronization allows user device 110 to share context updates made at user device 110 with other devices and allows user device 110 to receive context updates made at other devices so that correction suggestions can be made at user device 110 based on the user context determined at the other devices.

At step 308, user device 110 can receive user input specifying a word. For example, the user input can be textual input. The textual input can be input to a word processing document, instant message, email message, or any other textual input.

At step 310, user device 110 can generate a suggested replacement or correction for the specified word based on the context dictionary. For example, the application (e.g., application 118, application 120) that receives the user input can send the user input to suggestion client 112 on user device 110. Suggestion client 112 can send the user input to suggestion service 152 and suggestion service 152 can generate a correction suggestion based on the user’s context dictionary and send the correction suggestion to suggestion client 112. Alternatively, suggestion client 112 can generate the correction suggestion locally based on context dictionary 116. The suggestion client 112 can generate the correction suggestion by comparing the user input to context entries in context dictionary 116 and determining candidate alternative words in the context dictionary that might be appropriate replacements for the user input. Suggestion client 112 can then send the correction suggestion, including a selected candidate alternative word, to the application that received the user input.

At step 312, user device 110 can present the suggested replacement word on a display of user device 110. For example, the application that receives the correction suggestion can present the correction suggestion (e.g., word, graphic, etc.) on a display of the user device for the user to select. Upon receiving a user selection of the correction suggestion, the application can replace to user input (e.g., word) with the selected correction suggestion. Alternatively, the application that receives the correction suggestion can automatically replace the user input with the correction suggestion.

FIG. 4 is flow diagram of an example process 400 for disambiguating user input based on user context. For example, process 400 can be performed by user device 110 to disambiguate user input based on content that the user has recently selected to view.

At step 402, user device 110 can receive a selection of content. For example, user device 110 can receive user input selecting a webpage, contact information, word processing document, slideshow presentation, or any other type of content. The content can be selected by interacting with an application (e.g., web browser, address book application, word processing document, etc.) running on user device 110.

At step 404, user device 110 can generate a context dictionary based on the selected content. For example, the application (e.g., application 118, application 120) presenting the selected content can send the selected content to suggestion client 112. Suggestion client 112 can parse the selected content and generate context entries in context dictionary 116, as described above.

At optional step 406, user device 110 can synchronize the context dictionary with a second device. For example and as described above, user device 110 can synchronize context dictionary 116 with context dictionaries managed by user device 130 and/or server device 150. The synchronization allows user device 110 to share context updates made at user device 110 with other devices and allows user device 110 to receive context updates made at other devices so that correction suggestions can be made at user device 110 based on the user context determined at the other devices.

At step 408, user device 110 can receive a user query. For example, the user query can be a spoken command or request received or detected by an application (e.g., stand-alone application, operating system function or utility, system library, system API, or other hardware and/or software system, etc.) on user device 110. The user query can be a textual command or request received as user input by an application on user device 110.

At step **410**, user device **110** can determine that the user query is ambiguous. For example, when processing the user query, the receiving application can determine that there is more than one possible response to the user's query. For example, when the user's query poses the question "how are the Giants doing?" during an overlap between baseball and football seasons, the application can determine that the user may be asking about either the Giants baseball team or the Giants football team. When there is more than one possible response, the application can determine that the user's query is ambiguous.

At step **412**, user device **110** can resolve the ambiguity based on the context dictionary. For example, when the application that received the user's query determines that the query is ambiguous, the application can send the user's query (e.g., a textual representation of the user's query) to suggestion client **112**. Suggestion client **112** can compare the words in the user's query to context entries in context dictionary **116** to determine additional words that may clarify the user's query. For example, if the user was recently viewing a webpage discussing the Giants baseball team, then suggestion client **112** can add the word "baseball" to the user's query as a correction suggestion to disambiguate the user query. Suggestion client **112** can send the correction suggestion to the application.

At step **414**, user device **110** can perform a search based on the disambiguated user query. For example, the application can generate a response to the user's query "how are the Giants doing?" by performing an internet search using the query and the additional disambiguating word (e.g., correction suggestion) "baseball". The application can then receive search results related to the Giants baseball team.

At step **416**, user device **110** can present the search results. For example, the application on user device **110** can present the search results related to the Giants baseball team using audio data (e.g., speech synthesis) and/or using graphical data (e.g., text, images, etc.) presented on the display of user device **110**.

Graphical User Interfaces

This disclosure above describes various Graphical User Interfaces (GUIs) for implementing various features, processes or workflows. These GUIs can be presented on a variety of electronic devices including but not limited to laptop computers, desktop computers, computer terminals, television systems, tablet computers, e-book readers and smart phones. One or more of these electronic devices can include a touch-sensitive surface. The touch-sensitive surface can process multiple simultaneous points of input, including processing data related to the pressure, degree or position of each point of input. Such processing can facilitate gestures with multiple fingers, including pinching and swiping.

When the disclosure refers to "select" or "selecting" user interface elements in a GUI, these terms are understood to include clicking or "hovering" with a mouse or other input device over a user interface element, or touching, tapping or gesturing with one or more fingers or stylus on a user interface element. User interface elements can be virtual buttons, menus, selectors, switches, sliders, scrubbers, knobs, thumbnails, links, icons, radio buttons, checkboxes and any other mechanism for receiving input from, or providing feedback to a user.

Privacy

The present disclosure recognizes that the use of such personal information data, in the present technology, can be

used to the benefit of users. For example, the personal information data can be used to deliver targeted content that is of greater interest to the user. Accordingly, use of such personal information data enables calculated control of the delivered content. Further, other uses for personal information data that benefit the user are also contemplated by the present disclosure.

The present disclosure further contemplates that the entities responsible for the collection, analysis, disclosure, transfer, storage, or other use of such personal information data will comply with well-established privacy policies and/or privacy practices. In particular, such entities should implement and consistently use privacy policies and practices that are generally recognized as meeting or exceeding industry or governmental requirements for maintaining personal information data private and secure. For example, personal information from users should be collected for legitimate and reasonable uses of the entity and not shared or sold outside of those legitimate uses. Further, such collection should occur only after receiving the informed consent of the users. Additionally, such entities would take any needed steps for safeguarding and securing access to such personal information data and ensuring that others with access to the personal information data adhere to their privacy policies and procedures. Further, such entities can subject themselves to evaluation by third parties to certify their adherence to widely accepted privacy policies and practices.

Despite the foregoing, the present disclosure also contemplates embodiments in which users selectively block the use of, or access to, personal information data. That is, the present disclosure contemplates that hardware and/or software elements can be provided to prevent or block access to such personal information data. For example, in the case of advertisement delivery services, the present technology can be configured to allow users to select to "opt in" or "opt out" of participation in the collection of personal information data during registration for services. In another example, users can select not to provide location information for targeted content delivery services. In yet another example, users can select to not provide precise location information, but permit the transfer of location zone information.

Example System Architecture

FIG. **5** is a block diagram of an example computing device **500** that can implement the features and processes of FIGS. **1-4**. The computing device **500** can include a memory interface **502**, one or more data processors, image processors and/or central processing units **504**, and a peripherals interface **506**. The memory interface **502**, the one or more processors **504** and/or the peripherals interface **506** can be separate components or can be integrated in one or more integrated circuits. The various components in the computing device **500** can be coupled by one or more communication buses or signal lines.

Sensors, devices, and subsystems can be coupled to the peripherals interface **506** to facilitate multiple functionalities. For example, a motion sensor **510**, a light sensor **512**, and a proximity sensor **514** can be coupled to the peripherals interface **506** to facilitate orientation, lighting, and proximity functions. Other sensors **516** can also be connected to the peripherals interface **506**, such as a global navigation satellite system (GNSS) (e.g., GPS receiver), a temperature sensor, a biometric sensor, magnetometer or other sensing device, to facilitate related functionalities.

A camera subsystem **520** and an optical sensor **522**, e.g., a charged coupled device (CCD) or a complementary metal-

11

oxide semiconductor (CMOS) optical sensor, can be utilized to facilitate camera functions, such as recording photographs and video clips. The camera subsystem **520** and the optical sensor **522** can be used to collect images of a user to be used during authentication of a user, e.g., by performing facial recognition analysis.

Communication functions can be facilitated through one or more wireless communication subsystems **524**, which can include radio frequency receivers and transmitters and/or optical (e.g., infrared) receivers and transmitters. The specific design and implementation of the communication subsystem **524** can depend on the communication network(s) over which the computing device **500** is intended to operate. For example, the computing device **500** can include communication subsystems **524** designed to operate over a GSM network, a GPRS network, an EDGE network, a Wi-Fi or WiMax network, and a Bluetooth™ network. In particular, the wireless communication subsystems **524** can include hosting protocols such that the device **100** can be configured as a base station for other wireless devices.

An audio subsystem **526** can be coupled to a speaker **528** and a microphone **530** to facilitate voice-enabled functions, such as speaker recognition, voice replication, digital recording, and telephony functions. The audio subsystem **526** can be configured to facilitate processing voice commands, voiceprinting and voice authentication, for example.

The I/O subsystem **540** can include a touch-surface controller **542** and/or other input controller(s) **544**. The touch-surface controller **542** can be coupled to a touch surface **546**. The touch surface **546** and touch-surface controller **542** can, for example, detect contact and movement or break thereof using any of a plurality of touch sensitivity technologies, including but not limited to capacitive, resistive, infrared, and surface acoustic wave technologies, as well as other proximity sensor arrays or other elements for determining one or more points of contact with the touch surface **546**.

The other input controller(s) **544** can be coupled to other input/control devices **548**, such as one or more buttons, rocker switches, thumb-wheel, infrared port, USB port, and/or a pointer device such as a stylus. The one or more buttons (not shown) can include an up/down button for volume control of the speaker **528** and/or the microphone **530**.

In one implementation, a pressing of the button for a first duration can disengage a lock of the touch surface **546**; and a pressing of the button for a second duration that is longer than the first duration can turn power to the computing device **500** on or off. Pressing the button for a third duration can activate a voice control, or voice command, module that enables the user to speak commands into the microphone **530** to cause the device to execute the spoken command. The user can customize a functionality of one or more of the buttons. The touch surface **546** can, for example, also be used to implement virtual or soft buttons and/or a keyboard.

The memory interface **502** can be coupled to memory **550**. The memory **550** can include high-speed random access memory and/or non-volatile memory, such as one or more magnetic disk storage devices, one or more optical storage devices, and/or flash memory (e.g., NAND, NOR). The memory **550** can store an operating system **552**, such as Darwin, RTXC, LINUX, UNIX, MAC OS, iOS, WINDOWS, or an embedded operating system such as VxWorks.

The operating system **552** can include instructions for handling basic system services and for performing hardware dependent tasks. In some implementations, the operating system **552** can be a kernel (e.g., UNIX kernel). In some implementations, the operating system **552** can include

12

instructions for performing voice authentication. For example, operating system **552** can implement the contextual user input correction features as described with reference to FIGS. 1-4.

The memory **550** can also store communication instructions **554** to facilitate communicating with one or more additional devices, one or more computers and/or one or more servers. The memory **550** can include graphical user interface instructions **556** to facilitate graphic user interface processing; sensor processing instructions **558** to facilitate sensor-related processing and functions; phone instructions **560** to facilitate phone-related processes and functions; electronic messaging instructions **562** to facilitate electronic-messaging related processes and functions; web browsing instructions **564** to facilitate web browsing-related processes and functions; media processing instructions **566** to facilitate media processing-related processes and functions; GNSS/Navigation instructions **568** to facilitate GNSS and navigation-related processes and instructions; and/or camera instructions **570** to facilitate camera-related processes and functions.

The memory **550** can store other software instructions **572** to facilitate other processes and functions, such as the contextual user input correction processes and functions as described with reference to FIGS. 1-4.

The memory **550** can also store other software instructions **574**, such as web video instructions to facilitate web video-related processes and functions; and/or web shopping instructions to facilitate web shopping-related processes and functions. In some implementations, the media processing instructions **566** are divided into audio processing instructions and video processing instructions to facilitate audio processing-related processes and functions and video processing-related processes and functions, respectively.

Each of the above identified instructions and applications can correspond to a set of instructions for performing one or more functions described above. These instructions need not be implemented as separate software programs, procedures, or modules. The memory **550** can include additional instructions or fewer instructions. Furthermore, various functions of the computing device **500** can be implemented in hardware and/or in software, including in one or more signal processing and/or application specific integrated circuits.

What is claimed is:

1. A method comprising:

receiving, by a computing device, a first user input, in a first application, selecting a content item, the content item having a first contextual representation;

generating, by the computing device, one or more context entries in a context dictionary indicating the first contextual representation of the content item;

receiving, by the computing device, a second user input, in a second application different from the first application, specifying a properly spelled word, the properly spelled word having a second contextual representation different than the first contextual representation;

determining that the properly spelled word is a potential misspelling of a word associated with the first contextual representation based at least in part upon the one or more of the context entries in the context dictionary;

in response to determining that the properly spelled word is the potential misspelling of the word associated with the first contextual representation, generating, by the computing device, a suggestion for a replacement to the properly spelled word in the second application based on the context entries in the context dictionary, the

13

suggestion comprising an icon, a graphic, a second word, or a combination thereof having the first contextual representation;

causing, by the computing device, the suggestion to be presented on a display of the computing device; and 5

in response to receiving an indication to accept the suggestion, replacing the properly spelled word in the second application with the suggestion.

2. The method of claim 1, further comprising:

parsing the content item to determine one or more words 10 included in the content item; and

generating one or more context entries based on the one or more words in the content item.

3. The method of claim 1, further comprising:

determining that a particular content item corresponding 15 to a particular context entry in the context dictionary has been closed;

determining whether a period of time has elapsed since the particular content item was closed;

determining that the particular context entry is an active 20 context entry when the period of time has not elapsed since the particular content item was closed; and

generating, by the computing device, the suggestion for a replacement to the properly spelled word in the second application based on one or more active context entries 25 in the context dictionary.

4. The method of claim 3, wherein the active context entry comprises a context entry associated with a currently open content item, a content item closed within the period of time, or both. 30

5. The method of claim 3, further comprising:

determining that a particular content item corresponding to a particular context entry in the context dictionary has been closed;

determining whether a period of time has elapsed since 35 the particular content item was closed;

determining that the particular context entry is an inactive context entry when the period of time has elapsed since the particular content item was closed; and

generating, by the computing device, the suggestion for a 40 replacement to the properly spelled word in the second application based on one or more active context entries and not one or more inactive context entries in the context dictionary.

6. The method of claim 1, further comprising: 45

synchronizing the context dictionary on the computing device with one or more context dictionaries on other user devices; and

generating, by the computing device, the suggestion for a 50 replacement to the properly spelled word in the second application based on the context entries in the context dictionary received from other user devices.

7. The method of claim 1, wherein generating, by the computing device, the suggestion for a replacement to the properly spelled word in the second application based on the 55 context entries in the context dictionary includes receiving the suggestion from a network server that includes a user context dictionary that has been synchronized with the context dictionary on the computing device.

8. The method of claim 1, further comprising: 60

comparing the properly spelled word to a context dictionary;

based on a comparison of the properly spelled word to the context dictionary, determining that the properly spelled word should be changed to the icon, the 65 graphic, the second word, or the combination thereof, indicated by the context dictionary; and

14

preventing the computing device from indicating that the properly spelled word should be changed to the icon, the graphic, the word, or the combination thereof when the comparison of the properly spelled word to the context dictionary indicates that a context of the properly spelled word matches at least one of the context entries in the context dictionary.

9. The method of claim 1 performed by a suggestion client, the suggestion client being different than the first application and the second application.

10. A non-transitory computer readable medium including one or more sequences of instructions that, when executed by one or more processors, cause the one or more processors to perform operations comprising:

receiving a first user input, in a first application, selecting a content item, the content item having a first contextual representation;

generating one or more context entries in a context dictionary indicating the first contextual representation of the content item;

receiving a second user input, in a second application different from the first application, specifying a properly spelled word, the properly spelled word having a second contextual representation different than the first contextual representation;

determining that the properly spelled word is a potential misspelling of a word associated with the first contextual representation based at least in part upon the one or more of the context entries in the context dictionary;

in response to determining that the properly spelled word is the potential misspelling of the word associated with the first contextual representation, generating a suggestion for a replacement to the properly spelled word in the second application based on the context entries in the context dictionary, the suggestion comprising an icon, a graphic, a second word, or a combination thereof having the first contextual representation;

causing display of the suggestion; and

in response to receiving an indication to accept the suggestion, replicating the properly spelled word in the second application with the suggestion.

11. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions cause the one or more processors to perform operations comprising:

parsing the content item to determine one or more words indicating the first contextual meaning; and

generating one or more context entries in the context dictionary based on the one or more words in the content item.

12. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions cause the one or more processors to perform operations comprising:

determining that a particular content item corresponding to a particular context entry in the context dictionary has been closed;

determining whether a period of time has elapsed since the particular content item was closed;

determining that the particular context entry is an active context entry when the period of time has not elapsed since the particular content item was closed; and

generating, by a computing device, the suggestion for a replacement to the properly spelled word in the second application based on the active context entries in the context dictionary.

13. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions cause the one or more processors to perform operations comprising:

15

determining that a particular content item corresponding to a particular context entry in the context dictionary has been closed;
 determining whether a period of time has elapsed since the particular content item was closed;
 determining that the particular context entry is an inactive context entry when the period of time has not elapsed since the particular content item was closed; and
 generating the suggestion for a replacement to the properly spelled word in the second application based on one or more active context entries and not one or more inactive context entries in the context dictionary.

14. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions cause the one or more processors to perform operations comprising:
 synchronizing the context dictionary on computing device with one or more context dictionaries on other user devices; and
 generating the suggestion for a replacement to the properly spelled word in the second application based on the context entries in the context dictionary received from other user devices.

15. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions that cause the one or more processors to perform an operation of generating a suggestion for a replacement to the properly spelled word in the second application based on the context entries in the context dictionary include instructions that cause the one or more processors to perform operations comprising accessing the suggestion from a network server that includes a user context dictionary that has been synchronized with the context dictionary on a computing device.

16. The non-transitory computer readable medium of claim 10, wherein the sequences of instructions cause the one or more processors to perform operations comprising:
 comparing the properly spelled word to a context dictionary; and
 based on a comparison of the properly spelled word to the context dictionary, determining that the properly spelled word is not in the context dictionary;
 determining that the properly spelled word is similar to one or more words of the context dictionary; and
 generating the suggestion for a replacement to the properly spelled word comprising the one or more words of the context dictionary that are similar to the properly spelled word.

17. A system comprising:
 one or more processors; and
 a non-transitory computer readable medium including one or more sequences of instructions that,
 when executed by the one or more processors, cause the one or more processors to perform operations comprising:
 receiving a first user input, in a first application, selecting a content item, the content item having a first contextual representation;
 generating one or more context entries in a context dictionary indicating the first contextual representation of the content item;
 receiving a second user input, in a second application different from the first application, specifying a properly spelled word, the properly spelled word having a second contextual representation different than the first contextual representation;
 determining that the properly spelled word is a potential misspelling of a word associated with the first

16

contextual representation based at least in part upon the one or more of the context entries in the context dictionary;

in response to determining that the properly spelled word is the potential misspelling of the word associated with the first contextual representation, generating a suggestion for a replacement to the properly spelled word in the second application based on the context entries in the context dictionary, the suggestion comprising an icon, a graphic, a second word, or a combination thereof having the first contextual representation;

causing the suggestion to be presented on a display of the system; and

in response to receiving an indication to accept the suggestion, replacing the properly spelled word in the second application with the suggestion.

18. The system of claim 17, wherein the one or more sequences of instructions cause the one or more processors to perform operations comprising:

parsing the content item to determine one or more words included in the content item; and

generating one or more context entries in the context dictionary based on the one or more words in the content item.

19. The system of claim 17, wherein the one or more sequences of instructions cause the one or more processors to perform operations comprising:

determining that a particular content item corresponding to a particular context entry in the context dictionary has been closed;

determining whether a period of time has elapsed since the particular content item was closed;

determining that the particular context entry is an active context entry when the period of time has not elapsed since the particular content item was closed; and

generating the suggestion for a replacement to the properly spelled word based on one or more active context entries in the context dictionary.

20. The system of claim 17, wherein the one or more sequences of instructions cause the one or more processors to perform operations comprising:

determining that a particular content item corresponding to a particular context entry in the context dictionary has been closed;

determining whether a period of time has elapsed since the particular content item was closed;

determining that the particular context entry is an inactive context entry when the period of time has not elapsed since the particular content item was closed; and

generating the suggestion for a replacement to the properly spelled word in the second application based on one or more active context entries and not one or more inactive context entries in the context dictionary.

21. The system of claim 17, wherein the one or more sequences of instructions cause the one or more processors to perform operations comprising:

synchronizing the context dictionary on the system with one or more context dictionaries on other user devices; and

generating the suggestion for a replacement to the properly spelled word based on the context entries in the context dictionary received from other user devices.

22. The system of claim 17, wherein the one or more sequences of instructions that cause the one or more processors to perform an operation of generating a suggestion for a replacement to the properly spelled word in the second

application based on the context entries in the context dictionary include instructions that cause the one or more processors to perform operations comprising receiving the suggestion for a replacement to the properly spelled word from a network server that includes a user context dictionary 5 that has been synchronized with the context dictionary on the system.

23. The system of claim **17**, wherein the first application is a web browser and the second application is a word processing application. 10

24. The system of claim **23**, wherein generating the suggestion for a replacement to the properly spelled word comprises:

comparing the properly spelled word to the context dictionary; 15

based on a comparison of the properly spelled word to the context dictionary, determining that the properly spelled word is not in the context dictionary;

determining that the properly spelled word is similar to one or more words of the context dictionary; and 20

generating the suggestion for a replacement to the properly spelled word comprising the one or more words of the context dictionary that are similar to the properly spelled word.

* * * * *

25